Hybrid Sankey diagrams: visual analysis of multidimensional data for understanding resource use

R.C. Lupton[a,*], J.M. Allwood[a]

[a]*Department of Engineering, University of Cambridge, Trumpington St, Cambridge, CB2 1PZ, UK*

Abstract

Sankey diagrams are used to visualise flows of materials and energy in many applications, to aid understanding of losses and inefficiencies, to map out production processes, and to give a sense of scale across a system. As available data and models become increasingly complex and detailed, new types of visualisation may be needed. For example, when looking for opportunities to reduce steel scrap through supply chain integration, it is not enough to consider simply flows of "steel" — the alloy, thickness, coating and forming history of the metal can be critical. This paper combines data-visualisation techniques with the traditional Sankey diagram to propose a new type of "hybrid" Sankey diagram, which is better able to visualise these different aspects of flows.

There is more than one way to visualise a dataset as a Sankey diagram, and different ways are appropriate in different situations. To facilitate this, a systematic method is presented for generating different hybrid Sankey diagrams from a dataset, with an accompanying open-source Python implementation. A common data structure for flow data is defined, through which this method can be used to generate Sankey diagrams from different data sources such as material flow analysis, life-cycle inventories, or directly measured data. The approach is introduced with a series of visual examples, and applied to a real database of global steel flows.

Keywords: Sankey diagram, hybrid Sankey diagram, data cubes, star schema, Material Flow Analysis, visualisation

1. Introduction

Sankey diagrams are used to visualise flows of energy, materials or other resources in a variety of applications. Schmidt (2008a) reviewed the history and uses of these diagrams. Originally, they were used to show flows of energy, first in steam engines, more recently for modern systems such as power plants (e.g. Giuffrida et al., 2011) and also to give a big-picture view of global energy use (Cullen and Allwood, 2010). As well as energy, Sankey diagrams are widely used to show flows of resources (Schmidt, 2008a). Recent examples in this journal include global flows of tungsten (Leal-Ayala et al., 2015), biomass in Austria (Kalt, 2015), and the life-cycle of car components (Diener and Tillman, 2016). More widely, they have been used to show global production and use of steel and aluminium (Cullen et al., 2012; Cullen and Allwood, 2013), and flows of natural resources such as water (Curmi et al., 2013). In all of these cases, the essential features are: (1) the diagram represents physical flows, related to a given functional unit or period of time; and (2)

*Corresponding author. Tel: +44 1223 332876
Email address: rcl33@cam.ac.uk (R.C. Lupton)

the magnitude of flows is shown by the link[1] widths, which are proportional to an extensive property of the flow such as mass or energy (Schmidt, 2008b). Creating these diagrams is supported by software tools such as e!Sankey (ifu Hamburg, 2017), and several Life-Cycle Assessment (LCA) and Material Flow Analysis (MFA) packages include features to create Sankey diagrams.

Separately from these applications, in the world of data visualisation, Sankey diagrams are used to visualise arbitrary multidimensional data in quite a different way. For example, in a multidimensional dataset describing passengers on the Titanic (Kosara et al., 2006), with dimensions including ticket class and gender, the width of the link between *3rd class* and *male* shows how many of the 3rd class passengers were male. Although these "datavis" diagrams are visually similar to the "traditional" Sankey diagrams, with link widths being using to display information, it is the relationship between different attributes of the dataset which is shown rather than any physical flow. The nodes in a datavis diagram represent categories, not "real" processes where one flow ends and another starts. This type of visualisation was formalised by Kosara et al. (2006) as Parallel Sets (although they did not refer to it as a "Sankey diagram") and developed in tools including Fineo (Density Design, 2010). In engineering applications, this type of Sankey diagram has been used by Bajželj et al. (2013) to visualise global greenhouse gas emissions, allocated according to different dimensions such as *sector* and *fuel*. Konadu et al. (2015) visualised UK land area in this way, broken down into categories relevant to assessing biofuel production potential.

Both these types of Sankey diagram have proved useful. They look deceptively similar to each other, but they use the same visual structure to mean quite different things. This paper explores how these two types of Sankey diagram can be united, to develop a new "hybrid" Sankey diagram.

The new visualisations made possible by doing this are becoming particularly relevant as available data is becoming increasingly detailed. For example, Flint et al. (in preparation) constructed an MFA of flat steel production and manufacturing in the EU from a commercial sales database. When looking for opportunities to reduce steel scrap through supply chain integration, it is not enough to consider simply flows of "steel" — the alloy, thickness, coating and forming history of the metal can be critical. In a traditional Sankey diagram, material types are distinguished by colour or shading (Schmidt, 2008b), but in this case treating all these combinations as distinct material types would lead to far too many categories to be useful. A hybrid Sankey diagram deals directly with the multidimensional nature of the data, for example to show how different sectors use different thicknesses of steel, and how much of each is controlled via direct sales or via distributors. This remains embedded within a traditional Sankey diagram showing real flows and real processes, with mass conserved throughout.

There is more than one way to visualise a given dataset as a Sankey diagram, and different ways are appropriate in different situations (Kopec, 2015). Nuttbohm et al. (2009) discuss how Sankey diagrams are most effective when their structure matches the understanding and ideas of the target viewer, and Soundararajan et al. (2014) give examples of how different Sankey diagrams of national energy systems are suited to different objectives. Riehmann et al. (2005) and Alemasoom et al. (2015) argued that it should be easy to adjust the level of detail in a Sankey diagram. The approach presented in this paper facilitates this by developing a systematic method for generating different hybrid Sankey diagrams from a dataset. This is implemented in an open-source Python package which accompanies the paper.

The data shown using Sankey diagrams can come from several sources, including MFA or LCA models, directly measured data, and published statistics. The tools developed in this paper will be most useful if they work with data from any of these sources. Each uses different concepts and terminology, so the first part of the paper (Section 2) defines a common data structure, which is not tied to any particular modelling method

[1]The lines of a Sankey diagram are variously called links, streams or arcs; here we use "links".

Table 1: Terminology used by different modelling approaches to describe systems of flows

Approach	Flows between…	Notes
Process engineering[1]	*process units*	Process units include transformation & accumulation
MFA[2]	*processes*	Processes include transformation, transportation & storage
MFA[3]	*processes* & *stocks*	Flows connect processes (transformation) to stocks (storage)
SFA[4]	*cells*	Storage in cells; no transformation in SFA
MFN[5]	*transitions* & *places*	Flows connect transitions (transformation) to places (storage/connection)

MFA = Material Flow Analysis, SFA = Substance Flow Analysis, MFN = Material Flow Networks (Petri nets).
Example references: 1: Narasimhan and Jordache (1999). 2: Brunner and Rechberger (2003). 3: Geyer et al. (2007). 4: Löfving et al. (2006). 5: Möller et al. (2000).

or data source, that can be used as the starting point for creating visualisations. In the second part of the paper, Sections 3 and 4 develop a systematic method for describing and creating hybrid Sankey diagrams based on this common data structure. Section 5 gives details of the open-source Python implementation, and applies it to a real example dataset. Finally, Section 6 discusses the applicability of this approach to different data sources and styles of Sankey diagram, and suggests possible future work.

2. Common database structure

This section starts by briefly reviewing the different concepts and terminology used by different modelling approaches, to inform the choice of a common data structure for representing the flow data to be shown in the Sankey diagram. *Data cubes* are then introduced, as a useful database structure for multi-dimensional data. The aggregation operations are then described which are later used in the creation of Sankey diagrams from the data.

2.1. Concepts and terminology

The concepts and terminology used by several modelling methods are summarised in Table 1. In each case, *flows* represent movement of material or energy, associated with a given time interval or functional unit, but the nature of the endpoints of the flow varies. Some approaches enforce a strict distinction between stocks & processes, or places & transitions, while others do not. To be as general as possible, this distinction is not enforced in the common data structure defined here, and in this paper the endpoints of flows are referred to as "processes" that can include transformation, transportation, and/or storage (the definition given by Brunner and Rechberger, 2003).

Although inclusion of stock levels is not strictly necessary for production of Sankey diagrams (which show only the flows), they are included in the data structure along with the flows as they are important for giving a complete picture of the system over time. The stock-level data could be exploited in future by different visualisations based on the same data structure.

Conservation of mass or energy should be satisfied by the flows in the data structure. The requirements for this are as follows. For simplicity, in the following flows are assumed to be quantified by their mass, but the substitution of energy or any other conserved quantity should be straightforward.

Each flow can be written as f_{jkmt}, where the flow is of material m, from process j to process k, over time interval t. The stock level of material m in process k at the start of time interval t is s_{kmt}. Here "material" means an identifier which distinguishes types of flow at the finest level of detail that is of interest.

Possible values are therefore not just "steel" or "wood", but also "3mm thick uncoated steel" and "4mm thick galvanised steel", if that is a useful distinction to make in a particular application. For lack of a better word, non-material flows such as electricity are also included in the definition of "material".

Mass is allowed to accumulate in processes which include stocks. In time interval t, the accumulation of stock of material m is Δs_{kmt}, and stock levels can therefore be calculated as

$$s_{km(t+1)} = s_{kmt} + \Delta s_{kmt} \tag{1}$$

Conservations of mass must apply at all processes. At process k, net inflow is balanced by net destruction of material and accumulation of stock:

$$\sum_j f_{jkmt} - \sum_l f_{klmt} + c_{kmt} - d_{kmt} - \Delta s_{kmt} = 0 \qquad \text{for all } t, m \tag{2}$$

where c_{kmt} and d_{kmt} are the mass of type m created and destroyed in process k during time interval t. Creation and destruction of mass may only occur during transformation from one material type to another, so the net creation must be zero summed over all materials:

$$\sum_m [c_{kmt} - d_{kmt}] = 0 \qquad \text{for all } t, k \tag{3}$$

Combining Equations (2) and (3) for a pure process ($\Delta s = 0$) shows that inflows and outflows must balance:

$$\sum_m \left[\sum_j f_{jkmt} - \sum_l f_{klmt} \right] = 0 \qquad \text{for all } t, \text{ for pure process } k \tag{4}$$

Transportation can be described in the same terms. A simple transfer between two stocks in different locations can be represented by a flow directly connecting them. If there are transmission losses, or there is more than one source or target stock, an intermediate transportation process is needed.

2.2. Data cubes

A data cube is a way of representing data as a set of points in multidimensional space. Each point is called a *fact*, and has some associated quantitative *measures*. In the classic example of a sales database, the dimensions might be time interval, shop branch, and product type, with measures for quantity and sales; this allows queries such as "what are the total sales for 2011, grouped by shop branch?" to be answered. More complex queries are possible because the database can store additional information (metadata) about each of the dimensions. For example, adding details of the branches and dates would allow the database to answer the query "what are the total sales for Mondays in 2011, grouped by shop size and country?".

Löfving et al. (2006) used data cubes to manage the data for a Substance Flow Analysis (SFA). Each flow is viewed as a point in a multi-dimensional cube, whose dimensions correspond to the time interval, source, target, and material type of the flow. This structure makes it possible to pick out and aggregate flows over different time intervals and subsystems. However, they did not make use of the capability of data cubes to associate additional metadata attributes with the dimensions.

There are many implementations of data cubes (Pedersen et al., 2001) but the simplest, the *star schema* database (Kimball, 1996), is sufficient for the present application. In a star schema database, the facts are stored as rows in a *fact table*, with columns corresponding to the dimensions and measures. *Dimension tables* store additional metadata about each dimension value that appears in the fact table. There is a large amount of literature on OLAP (On-Line Analytical Processing) databases (Chaudhuri and Dayal, 1997),

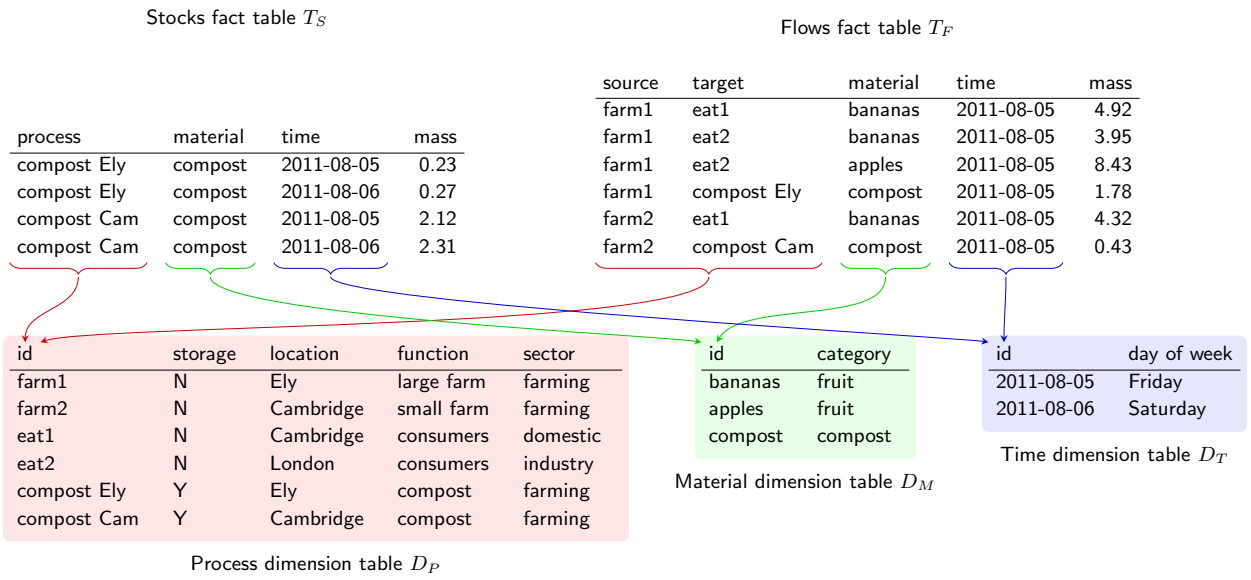


Figure 1: Extract of the example star schema database tables for flow and stock data.

which is concerned with the efficient implementation of data cubes over large datasets. For this paper, implementation details are not a primary concern, but it is worth noting that the data cube approach is successfully used at large scale. There is also a line of research generalising OLAP to work with graphs rather than single facts, known as Graph OLAP (Chen et al., 2009). Although this sounds relevant, the extra complexity is not necessary for the present case. It would become relevant if graph-like queries, such as finding shortest paths through the network, were of interest.

The star schema is used here as the basis of the common data structure described above. There are two fact tables T_F and T_S containing the flows f_{jkmt} and the stock levels s_{kmt}. There are three dimension tables, for the processes j & k, the material types m and the time intervals t, describing additional attributes which will depend on the application. Examples of material attributes are composition, temperature, and owner; for time intervals: day of week or weather; and for processes: location, owner, or function. Figure 1 shows an extract of an example database describing flows of fruit from farms to consumers. For generality, all processes are treated equally in the database, but it may be useful to track whether they include storage or not. The "storage" flag shown in the process dimension table D_P may be used to render storage and non-storage processes differently in the final Sankey diagram.

2.3. Aggregation

In general the values in the database are not used directly, but are aggregated in various ways before visualisation. To do this, first the relevant data must be *selected*, then *partitioned* into the categories of interest, before actually *aggregating* the data within each category. These steps are outlined in this section, with more details given in Appendix B.

Data points are selected primarily based on the process(es) they relate to. Sometimes the selection may be narrowed to include only certain material types or time intervals. For example, to pick out data describing some of the outputs of `farm1` (Figure 1), the following selections could be defined:

$$S_{\text{source}} = \{\text{farm1}\}$$
$$S_{\text{target}} = \{\text{eat1}, \text{eat2}, \text{compost Ely}\}$$

where S stands for *selection*. If applied to the flow table T_F shown in Figure 1, these selections would select the first 4 flows in the table. Although for this example the process ids are explicitly listed, in practice the database also allows selection based on metadata in the process, material and time dimension tables.

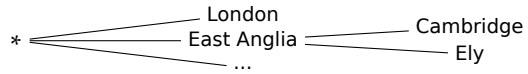
Once the required data points have been selected, they are partitioned into subgroups. Mathematically the subgroups are defined by a partition function P, which maps facts in the flow or stock fact tables onto labels L_P. The simplest partition, P^*, maps all facts to the same label; that is, it aggregates everything together. It is common to group flows and stocks by the values of some attribute, written $P(\textit{attribute})$. In the example above, the partition P(material.category) would map flows 1, 2, and 3 to the label "fruit", and map flow 4 to the label "compost". Although these are the most common ways to define them, partitions can be arbitrarily complex. In general, flows can be partitioned according to their four dimensions — source, target, material, and time — while stock levels can be partitioned according to their three dimensions.

The final step is to aggregate the data points in each subgroup. Extensive quantities that will be visualised by the width of the links in the Sankey diagram (e.g. mass or energy) must be summed. In the example above, the result is two subgroups of mass flow 17.30 and 1.78 respectively. Additional intensive or extensive measures may be combined by any appropriate aggregation function, such as calculating the mean or maximum temperature of the flows in the subgroup.

2.4. Hierarchies

Although the aggregation procedure described above is sufficient, in practice it quickly becomes cumbersome to list the full set of attribute values when defining selections and partitions. Hierarchies can organise the metadata to make this easier.

A hierarchy $\mathcal{H}$ is associated with a column of the dimension table and is represented by a tree whose leaves are attribute values in the dimension table. For example, we might define a hierarchy $\mathcal{H}_{\text{loc}}$ based on the "location" column of the process dimension table D_P (Figure 1), with the following tree:



Using this, we can easily select processes in "East Anglia", say:

$$\begin{aligned} &S\left(\mathcal{H}_{\text{loc}}, \text{East Anglia}\right) \\ &\quad = \left\{k \,\middle|\, k \in D_P,\ \text{location}(k) \in \{\text{Cambridge}, \text{Ely}\}\right\} \\ &\quad = \{\text{farm1}, \text{farm2}, \text{eat1}, \text{compost Cam}, \text{compost Ely}, \ldots\} \end{aligned}$$

In general, the selection is given by

$$S\left(\mathcal{H}_i, n\right) = \left\{k \,|\, k \in D_i, C_i(k) \in \text{leaves}_i(n)\right\} \tag{5}$$

where D_i and C_i are the dimension table and attribute column associated with hierarchy $\mathcal{H}_i$, and $\text{leaves}_i(n)$ is the set of leaves below n in the tree. Note that multiple hierarchies can be associated with the same attribute, such as "year–month–day" and "week–day" for time, or geographical and political boundaries for location.

The same hierarchies can be used to define partitions by describing the subgroups in terms of points in the tree rather than directly in terms of attribute values in the dimension tables. Given a list of points in the tree n_j to act as subgroups, a hierarchy $\mathcal{H}_i$ defines a partition

$$P\left(\mathcal{H}_i, \left\{n_j\right\}\right): \quad k \mapsto \begin{cases} n_j & \text{if } k \in S\left(\mathcal{H}_i, n_j\right) \text{ for all } n_j \\ \text{"other"} & \text{otherwise} \end{cases} \tag{6}$$

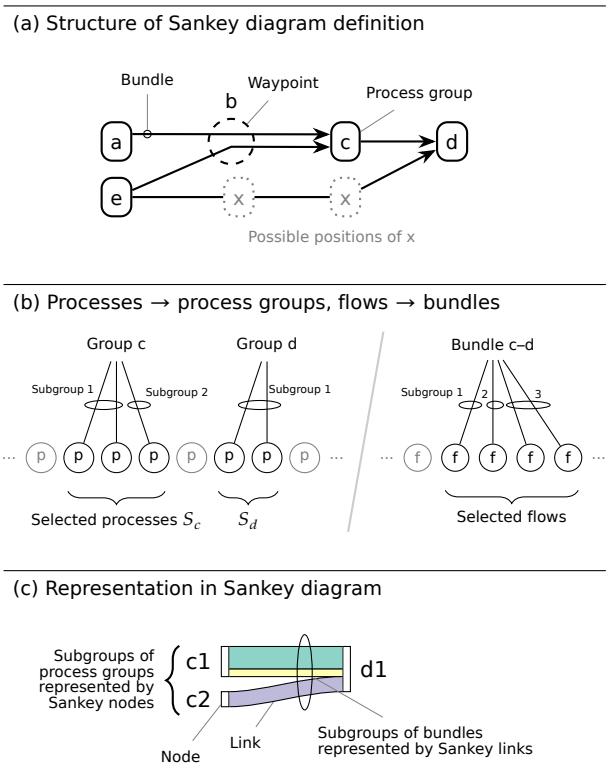


Figure 2: The Sankey diagram definition. (a) The structure of the Sankey diagram is defined by bundles, process groups and waypoints. Their arrangement is significant: for example, there are two possible positions for process group x, which could be aligned with b or c. (b) Processes are combined into process groups, and flows are combined into bundles. The processes and flows can be partitioned in various ways into subgroups. (c) The subgroups are represented visually by nodes and links in the Sankey diagram.

3. Specifying the desired diagram

There are very many possible ways to visualise the same dataset as different hybrid Sankey diagrams, so some means is needed of specifying a particular diagram. A "Sankey diagram definition" (SDD) acts as a specification, which can in principle be shared between multiple datasets, describing a particular way of presenting the flow data in the dataset. At the highest level, it defines the structure of the diagram: which processes and flows in the underlying database appear in the diagram, and their relative arrangement (Figure 2a). This is effectively a description of a traditional Sankey diagram structure. Within this structure, the flows can be presented at different levels of detail (Figure 2b), including datavis-style aggregation based on multiple attributes. This logical structure is then represented visually by nodes and links in the Sankey diagram (Figure 2c).

The SDD is introduced in this section through a series of examples, using the same example data shown in Figure 1, before giving a more formal description of the SDD. Examples of the Sankey diagrams resulting from the SDDs are given throughout; the procedure which produces them will be discussed in Section 4.

3.1. Diagram structure

The diagram structure is defined by combining three elements. *Process groups* (shown by solid boxes in Figure 2a) represent a set of processes in the underlying database, defined by a selection S_j (Section 2.3).

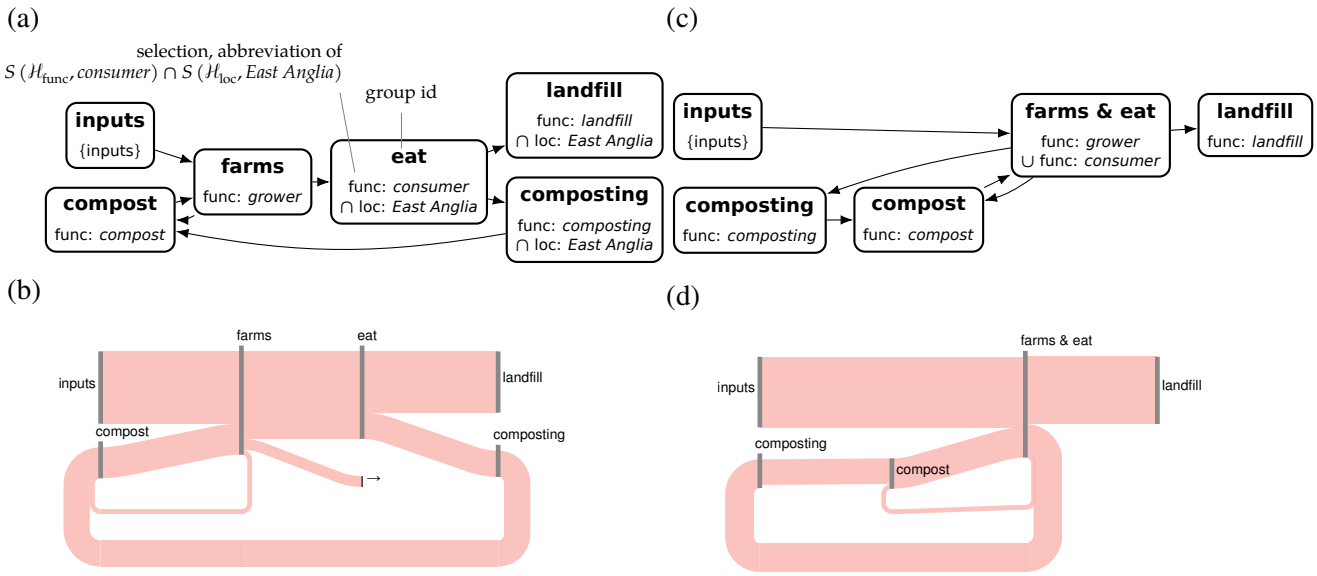
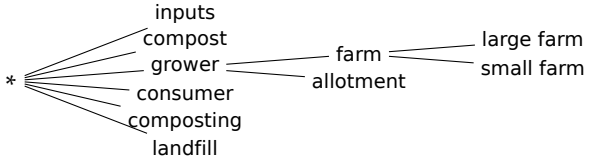


Figure 3: Structure of the SDD. (a) Graphical representation of an SDD. Each group shows its id and selection (see Section 2.3); the horizontal position of the process groups indicates the layers, and the vertical ordering shows the order within the layer. (b) The Sankey diagram resulting from the definition above. Note that the "farms" process group includes farms in all locations, whereas the selection for the "eat" process group includes consumers only in East Anglia (as defined in Section 2.4). This means that the link from "farms" to "eat" does not show all of the mass leaving the farms, and the remaining mass must be shown in some other way: the automatically-generated link labelled "from farms" in the Sankey diagram. If the automatic link is not suitable, it can be defined explicitly as shown later in Section 3.3. (c) An alternative SDD, with the (d) resulting Sankey diagram, showing the same data in a different way.

Bundles (arrows in Figure 2a) represent sets of underlying flows between process groups. *Waypoints* (dashed circle in Figure 2a) are used to gain more control over the layout of the diagram, and to reaggregate the same flows in multiple ways to show relationships between attributes, as in a datavis-style Sankey diagram. Unlike process groups, they do not represent underlying processes.

The layout of the diagram is defined by the arrangement of the process groups and waypoints. In a simple diagram there is usually an obvious arrangement, but in the presence of loops, there are more possibilities. Similarly, if there are multiple paths of different lengths between two processes, there is slack allowing a choice of the horizontal position of process groups in the shorter path (see example in Figure 2a). It is important that the arrangement is explicit, since the vertical slices through the Sankey diagram often have an associated meaning. However, the SDD should not involve exact coordinates, since it exists independently of any particular sets of flows, and the correct coordinates are not known until the widths of the flows is known. The SDD therefore contains an *ordering* which organises the node groups and waypoints into *layers*. Section 6.3 will discuss how this generalises to alternative visual styles of Sankey diagram.

An example SDD is given in Figure 3a. As well as the previously-defined $\mathcal{H}_{\text{loc}}$ hierarchy, the selections make use of a "function" hierarchy $\mathcal{H}_{\text{func}}$ as folows:



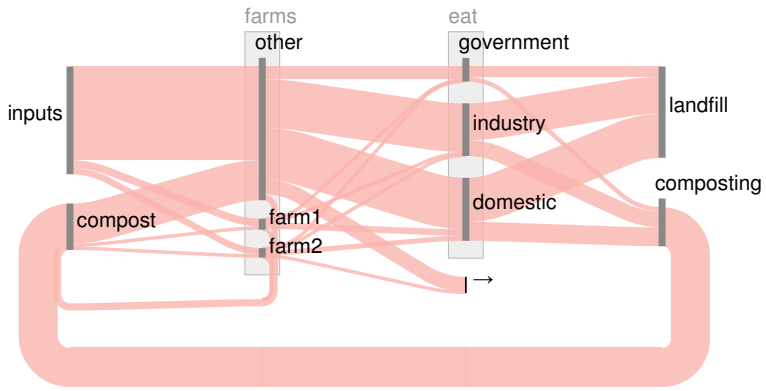


Figure 4: A Sankey diagram with the same structure as in Figure 3b but partitioned differently. The "farms" process group has a bespoke partition, based on the id of the underlying processes: [`farm1`] → "farm1", [`farm2`] → "farm2", [`farm3`, `farm4`, ...] → "other". On the other hand, the "eat" process group has the attribute-based partition P(process.sector).

The resulting Sankey diagram is shown in Figure 3b, including an additional link to satisfy conservation of mass (see caption). Figure 3c-d shows that the same data can be easily presented in a different form.

3.2. Aggregation of processes and flows

Within the high-level structure of the diagram, the level of detail is determined by partitioning the real processes and flows into subgroups (Figure 2b), which become the nodes and links of the Sankey diagram (Figure 2c). There are three places within a hybrid Sankey diagram that this partitioning happens. Firstly, the processes within a process group can be partitioned into subgroups reflecting different categories. Secondly, additional nodes can be added representing different sets of categories, to "reaggregate" the flows and show the relationships between attributes. Finally, the flows themselves can be partitioned, to distinguish different material types using different coloured links in the diagram. These three types of partitioning can be described by the same building blocks of process groups, waypoints and bundles.

Partitioning groups of processes

The level of detail at which processes are shown is controlled by defining a partition for the process group (Section 2.3). In the previous examples in Figure 3, all the processes in each group were lumped together, to show only a single node in the final Sankey diagram for each process group in the SDD; this corresponds to the "all" partition P^*. Figure 4 shows the effect of a different partition of the `farms` and `eat` process groups: although the basic structure of the diagram is the same as in Figure 3b, additional nodes and links have been created to show the subgroups.

Reaggregation

To explore the relationships between different attributes of flows and processes, a set of flows can be *reaggregated* by passing a single bundle through several waypoints, each with a different partition: at each point, the partition defined at the waypoint determines the aggregation of the flows in the bundle. Figure 5 illustrates the resulting datavis-style Sankey diagram. Because Figure 5 includes only one bundle, the same underlying "real" flows are shown across the whole diagram, only grouped in different ways. Later examples show how this datavis-style reaggregation can be embedded in a network of flows and processes, to give a hybrid Sankey diagram.

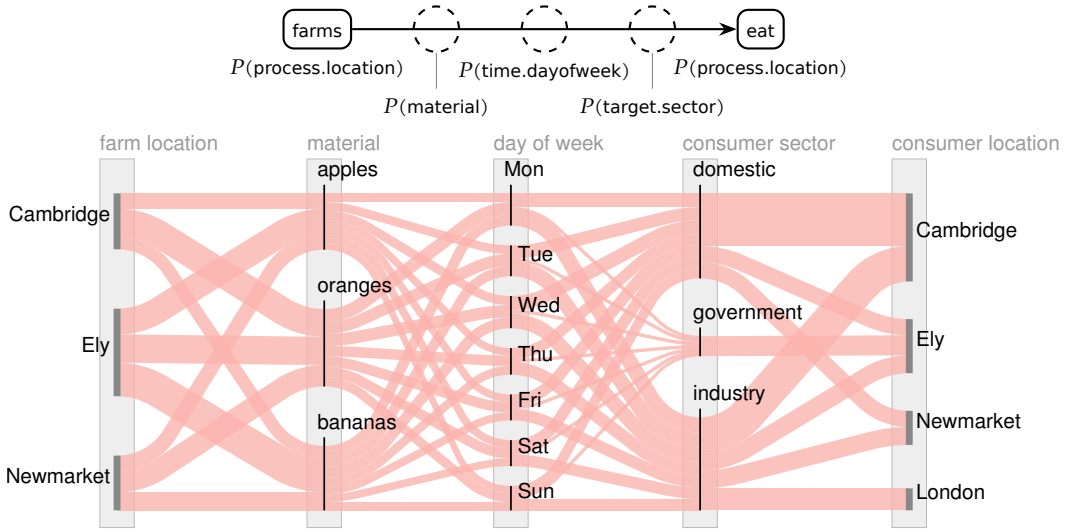


Figure 5: Waypoints add additional layers to compare multiple attributes. Any attribute of the flows — whether of the source process, target process, material type, or time interval — can be used. The Sankey definition above shows one bundle, between the process groups "farms" and "eat", passing through three waypoints. The partitions are shown below the process groups and waypoints (see Section 2.3). The resulting Sankey diagram shows, for example, that (in this unrealistic example) all consumers in London are part of the "industry" sector, flows of fruit are fairly even across the week, and the most popular fruit grown in Cambridge is oranges.

Partitioning bundles of flows

So far different types of flow have not been distinguished in the diagram: all the underlying flows between the same subgroups have been aggregated together. Instead, the flows selected by a bundle can be partitioned in a similar manner to the processes within a process group. Figure 6 shows an example of this.

As well as the categorical colour scale shown here, quantitative colour scales can also be used to indicate the value of some measure of the flows (e.g. flow temperature, or material cost).

3.3. Adjusting the appearance of the Sankey diagram

There are a number of further adjustments that can be made to the structure and appearance of the Sankey diagram using the same building blocks of the SDD. Figure 7 shows a more complex example and describes four adjustments.

3.4. Summary

In this section the concepts needed to define the structure of a particular desired hybrid Sankey diagram have been introduced. The definition $D = (N_G, N_W, B, L)$ consists of: a set of *process groups* N_G representing processes in the underlying database; a set of *waypoints* N_W which provide additional structure to the diagram; a set of *bundles* B, representing flows between processes; and an *ordering* L, defining the relative placement of the process groups and waypoints.

Each process group and waypoint j has a direction (left or right), a partition P_j (see Section 2.3), and a title which appears in the final diagram. The process groups N_G additionally have a selection S_j of underlying processes. Selections must not overlap between the process groups.

The bundles in B can be written as b^W_{jk}, which indicates a bundle b with source $j \in N_G$, target $k \in N_G$ and a set of waypoints $W \subset N_W$.

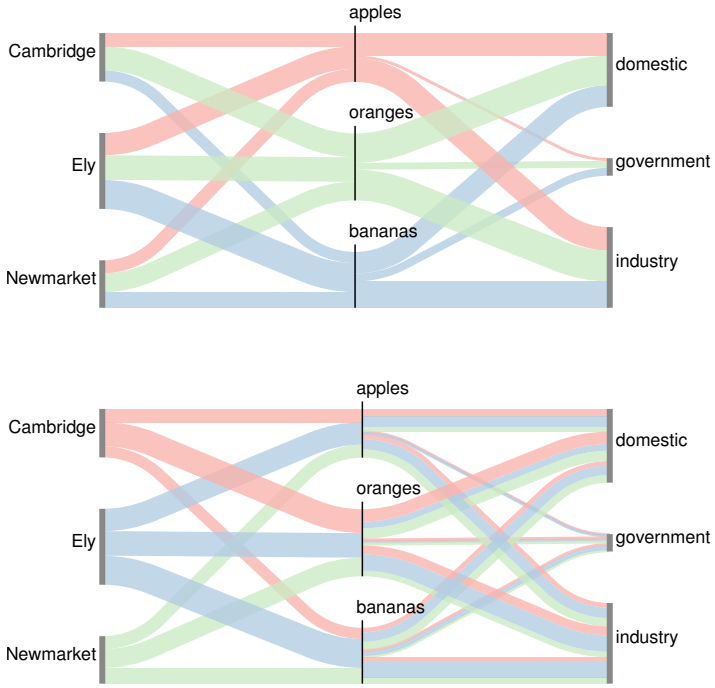


Figure 6: A similar diagram to Figure 5, but now partitioning bundles based on different attributes. These can be either attributes of the flow (`material`, above) or of the source or target processes (`source.location`, below).

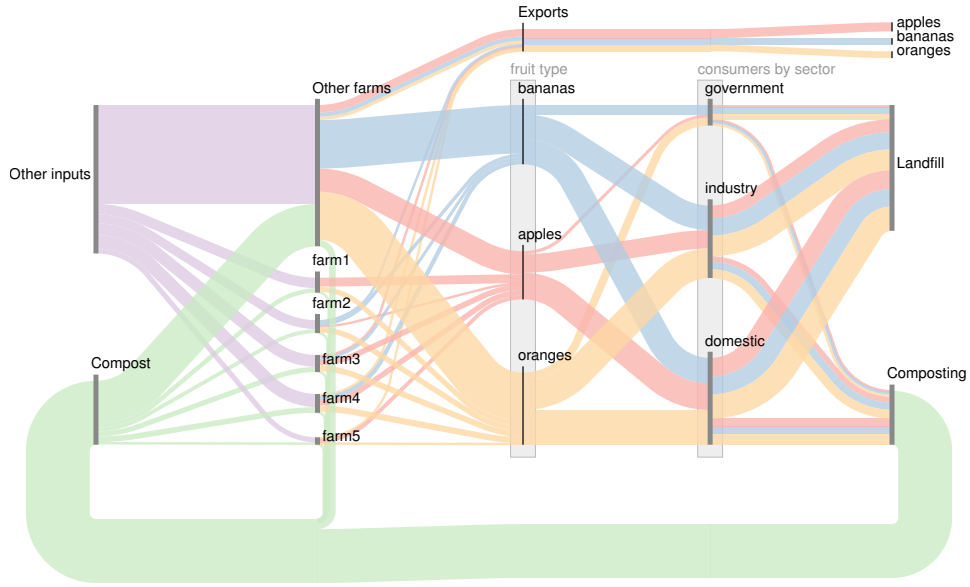


Figure 7: A more complex Sankey diagram. (1) The structure of the diagram can be simplified by merging flows from multiple bundles. Here, the return flows to "compost" are merged by defining a shared waypoint which both of the bundles pass through (as in Figure 2, top). (2) Import/export bundles are implicitly added to ensure conservation of mass, but by adding them explicitly their placement and partitioning can be controlled. Here, the export flows at the top of the diagram are partitioned by adding waypoints and an explicit export bundle "to elsewhere". (3) Sankey diagrams often naturally have horizontal "bands", such as the upper band containing the export flows, and the lower band containing the return flows. Bands can be defined as part of the ordering. (4) Although most of the flow is from left to right, some flows travel in the reverse direction. Here the shared return-flow waypoint described above is flowing from right to left.

The ordering consists of a set of layers, which are split into horizontal bands (see Figure 7):

$$L = \begin{bmatrix} L_{11} & L_{21} & \dots \\ L_{12} & L_{22} & \dots \\ \vdots & \vdots & \end{bmatrix} \tag{7}$$

where $L_{ij} \subset (N_G \cup N_W)$ is a list of process groups and waypoints in layer i and band j.

4. Aggregating flows into Sankey diagrams

Having chosen a particular view of the flow data by specifying an SDD, the Sankey diagram is actually created from the database in three steps: augmenting the SDD to ensure conservation of mass/energy is satisfied, routing flows across the diagram to create the final structure of links and nodes, and querying the database to find the size of each final link.

4.1. Ensuring conservation of mass/energy

For consistency of the final diagram, all flows to or from a process should be included, even if the target/source of the flow is not present in the diagram. The SDD (N_G, N_W, B, L) is therefore augmented with additional bundles B':

$$B' = \{b_{n\boxtimes} \,|\, n \in N_G, b_{n\boxtimes} \notin B\} \bigcup \{b_{\boxtimes n} \,|\, n \in N_G, b_{\boxtimes n} \notin B\} \tag{8}$$

where $\boxtimes$ represents "elsewhere", a special process group containing all processes in the database which are not included in another process group in the SDD. These additional bundles may be interpreted as import/exports or losses, depending on why the target/source has not been included in the diagram.

The augmented definition $(N_G, N_W, B \cup B', L)$ ensures that all flows to/from all process groups in N_G will be included in the final diagram. If all required bundles have been explicitly listed in B, then B' will be empty.

4.2. Final diagram structure

If the Sankey diagram is to be automatically laid out and drawn, its final structure must be determined by precisely routing any bundles whose endpoints/waypoints are not in adjacent layers. Sugiyama et al. (1981) give an algorithm for general layered graphs which involves adding "dummy nodes" to these "long" bundles. In the layered style of diagram drawn here, the nodes have an implied flow direction, so unlike a general layered graph, additional dummy nodes are needed in bundles connecting nodes of opposite direction. Details of the algorithm to build the final Sankey diagram structure are given in Appendix A. The result is represented by a layered graph $G_S = (V_S, E_S, L_S)$. As discussed in Section 6.3, this step is not necessary if the Sankey diagram is laid out by hand.

4.3. Flow values from database

The final step is to take the graph G_S describing the final structure of the diagram, and apply it to the underlying database to obtain actual values for the flows.

First, there may be a diagram-wide filter on the material types or time intervals to be shown, yielding a subset F' of the flow table T_F. Then the flows selected by each bundle are partitioned and aggregated to give the final links that appear in the diagram. Details of this process are given in Appendix B.

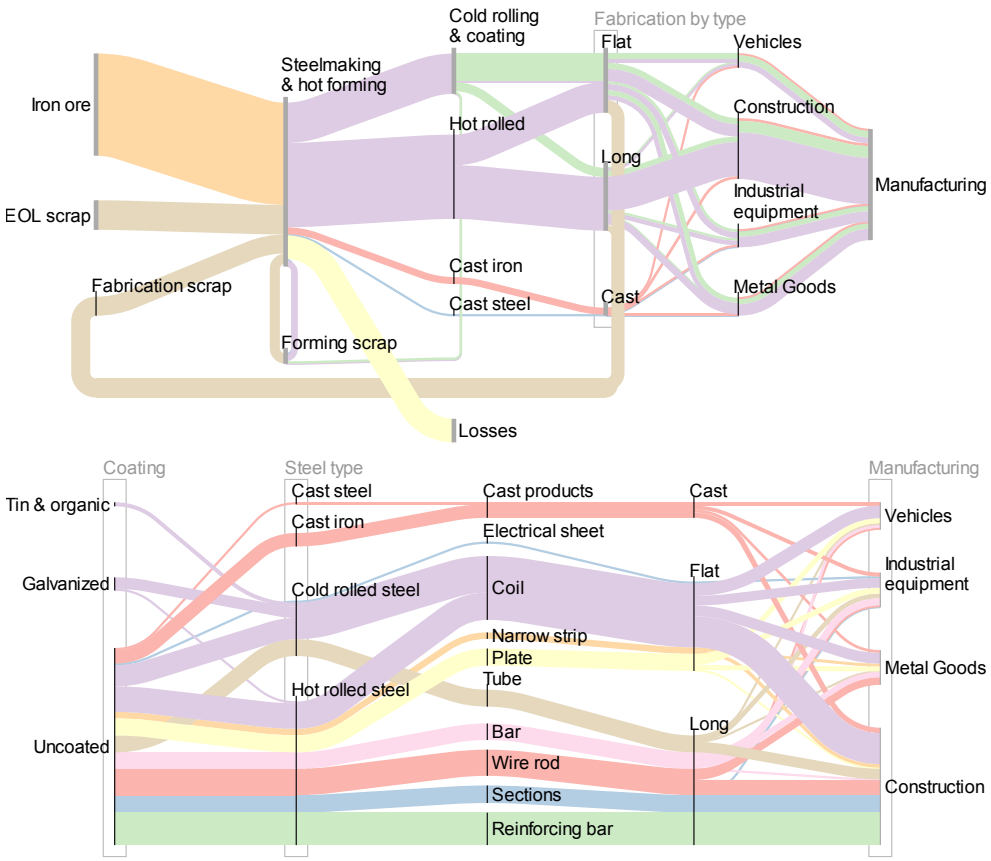


Figure 8: Cullen's global steel flow data, redrawn in an alternative form using the methodology of this paper. The same database underlies both diagrams; only the SDD is different. Above: a simplified diagram of similar structure to the original. The colours indicate the material type. Below: Flows of steel products from fabrication to manufacturing, partitioning by coating, steel type, shape and sector. These are the same set of flows seen at the right-hand side of the top diagram, but shown in more detail.

The result is contained in another layered graph, the results graph $G_R = (V_R, E_R, L_R)$, which contains all the information needed to actually draw the diagram. The nodes V_R correspond directly to nodes in the Sankey diagram. The edges $(j, k, m, t, x) \in E_R$ correspond to links in the Sankey diagram, where j and k are the source and target in V_R, m is the material, t is the time interval, and x is the link value. The ordering of the results graph L_R follows directly from the ordering of the previous graph L_S, given an ordering of the partition labels.

5. Implementation and applications

The method developed in this paper has been implemented as an open-source Python package (Lupton, 2016b). This package allows a SDD to be specified, and applied to a database of flows to produce the aggregated values needed to draw the final Sankey diagram. The diagrams in this paper are created from this information using another open-source package developed by the authors (Lupton, 2016a), which takes care of automatic placement of the nodes and layout of the links.

To demonstrate a more realistic application, beyond the made-up fruit data used so far, the values found by Cullen et al. (2012) for global flows of steel were put into the database format described in Section 2.2.

Figure 8 shows two Sankey diagrams generated from this database. These are two very different views of the same underlying data; without the methodology presented here, it would require a significant amount of manual data manipulation to get the data in the correct aggregated form.

The fruit and steel databases, and SDDs which produced all the Sankey diagrams in this paper are available in reproducible form online (Lupton and Allwood, 2016).

6. Discussion

In this paper a new hybrid Sankey diagram has been developed, which extends the traditional Sankey diagram better to show relationships in multidimensional data. The following sections discuss how well this approach generalises to different data sources, different Sankey diagram structures, and different visual styles, before concluding with suggestions for future work.

6.1. Applicability to different data sources

Beyond the made-up example data used in this paper, this approach has been applied by the authors and others to: a global MFA of steel (Section 5), a detailed MFA of steel sales in the EU (Flint et al., in preparation), measurements of energy and material flows at process and plant level in steelmaking (Gonzalez Hernandez et al., 2017), statistics on energy use (Paoli et al., 2017), and emissions of air pollutants in the UK (Mourão et al., in preparation). The requirements for structuring data in the way proposed here are quite low, and in all of these cases it was straightforward to do so. This approach has not yet been applied to data from life-cycle inventories or input-output models, but it should be applicable. Life-cycle inventories can include detailed descriptions of different types of goods and emissions, which might be well suited to this type of visualisation.

6.2. Applicability to different diagram structures

Figures 3–8 have given examples of the range of visualisations that are possible using hybrid Sankey diagrams. At the same time, this approach can also explain and generate some other published diagrams which do not quite fit into the traditional structure of "one flow, one link". The global steel Sankey diagram of Cullen et al. (2012) shows flows of 19 steel products to the 10 sectors that use them. To make the diagram clearer, these flows are grouped together first into 4 sector groups, which are then split into the full 10 sectors. This is not strictly a traditional Sankey diagram, because the sector group nodes only represent categories, not "real" processes. Viewed as a hybrid Sankey diagram, this can be naturally described and generated by placing a waypoint on the bundle from steel products to sectors (as in Figures 5–7). A further example of this type is the grouping of imported and domestic crops as "total crops" by Kalt (2015).

A great variety of Sankey diagrams have been published, and there will inevitably be some which do not fit into the theory and method described here. Nonetheless we believe that the structure behind a useful range of diagrams can be created in this way (including all of the diagrams reviewed by Schmidt (2008a,b), with the exception of stock inventories, which are not addressed by this paper, and perhaps the more unusual financial diagrams). However, the large variety of *visual styles* of Sankey diagrams is another matter, which is discussed next.

6.3. Different visual styles

There is a wide range of visual styles of Sankey diagram. Some resemble a single fat arrow with parts splitting and joining, while others show processes as rectangles with arrows going in many directions between them (Schmidt, 2008a). In our work (e.g. Cullen et al., 2012; Allwood et al.) we have tended to use

a simple visual style, in which there is a dominant flow direction left to right so that arrow heads on flows are not needed, and processes are shown by vertical lines, which tend to be arranged in vertical layers. This style is well-suited to presenting top-down analysis, where the majority of the flows move through different stages across the whole diagram. The arrangement of processes into vertical layers makes it possible to use layered graph layout algorithms to automatically construct the Sankey diagram (Lupton, 2016a). However, it tends to works less well for process-level flow-chart diagrams.

The examples in this paper, and the current versions of the open-source Python tools which accompany it, were developed in that context and reflect the style of our previous work. Nonetheless, readers who favour different styles could still benefit from the database structure, and the selection, partitioning and aggregation steps presented here. The result of these steps is the Sankey diagram graph structure, including the final width of each link, which can then be drawn as desired. Future development of the open-source tools to support other visual styles would be welcome. For example, modifying the SDD to include coordinates of process groups and waypoints would allow more flexible arrangements of the Sankey diagram nodes, at the expense of more effort in manual positioning.

6.4. Future work

There are three main areas of potential future work related to this approach: obtaining the detailed database of flow data in the first place, specifying the desired Sankey diagram, and developing alternative visualisations based on the same common data structure.

Some of the example Sankey diagrams given here (e.g. Figure 5) assume that a detailed dataset is available, disaggregated according to the relevant dimensions, but in many cases this may not be readily available. Lupton and Allwood (under review) show one way that this problem could be avoided, by accounting for the uncertainty in the disaggregated data that results from only some aggregated totals being known. This could allow the full disaggregated database to be constructed, albeit with potentially large uncertainty for many entries. The uncertainty in the resulting Sankey diagram will vary depending on how the data is subsequently aggregated for visualisation.

While the method presented here automates the preparation and aggregation of data for visualisation, it requires a precise definition of the desired result, and much of the effort involved in creating a Sankey diagram using this method goes into producing the SDD (Section 3). In future, it may be possible to develop higher-level interfaces to this definition, which could make the initial construction of a diagram quicker and easier, at the expense of less detailed control.

The approach described here could be extended to produce other visualisations, besides Sankey diagrams, based on the same common data structure. For example, suppose a pie chart is needed which breaks down the materials output from a production process to a certain consumer. The required aggregated data is the same as for a bundle with source process group containing the production process, target process group containing the consumer, and with a partition based on the material attributes of the flow. Similarly, visualisation of stock-level data has not been discussed here, but using similar database operations this data could be prepared for presentation in tables, as time series plots, or pie charts showing the breakdown at a point in time with respect to attributes of the process, material or time interval. More complex examples such as stock demographics diagrams (Cabrera Serrenho and Allwood, 2016) could also be generated by distinguishing cohorts of stock in the material dimension table.

The value and flexibility of this approach has hopefully been demonstrated through many examples of its use. We would like to encourage the publication and sharing of the results of resource analyses in a standard structure such as the one proposed here. By doing so, results would be made richer and more accessible to other researchers and interested viewers, who could then explore alternative views beyond those chosen by the authors.

Acknowledgements

This work was supported by EPSRC [EP/N02351x/1].

Appendix A. Building the final diagram structure

Algorithm 1 Building the view graph. The function dummies(x, y) gives the dummy nodes required to route the bundle from x to y, according to Figure 9. The new dummy nodes are inserted into the ordering L_S using a median-neighbour algorithm, similar to that described by Gansner et al. (1993), to minimise crossing flows. The problem is slightly different in that the original ordering given as part of the SDD should not be altered; only the ordering of the new dummy nodes within the fixed original nodes can be altered.

```
for all bundles b^W_jk in B ∪ B′ do
    nodes ← [j] + W + [k]
    for all x, y in PAIRWISE(nodes) do          ▷ PAIRWISE(1, 2, 3) gives (1, 2), (2, 3)
        if x ≠ ⊠ and y ≠ ⊠ then
            nodes' ← [x] + dummies(x, y) + [y]
            for all n in nodes' do
                add node n to V_S
                add node n to L_S               ▷ Position to minimise crossings of existing edges
            end for
            for all x′, y′ in PAIRWISE(nodes') do
                add edge (x′, y′, b^W_jk) to E
            end for
        end if
    end for
end for
```

Before the Sankey diagram can be drawn, the final structure of the diagram must be obtained by precisely routing any bundles which cross more than one layer. Sugiyama et al. (1981) give a method to do this for general layered graphs which involves adding "dummy nodes" to links whose source and target are not in adjacent layers. Because the nodes of a Sankey diagram have a flow direction, unlike a general layered graph, additional dummy nodes are needed in bundles connecting process groups of opposite direction. There are eight possible situations, depending on the relative placement and direction of the source and target. Figure 9 shows the outcome, assuming a convention that flows changing direction turn clockwise.

The final structure is represented by a layered graph $G_S = (V_S, E_S, L_S)$. Each edge in E_s is a tuple (v, w, b), where $v, w \in V_s$ are the source and target of the edge, and $b \in B \cup B'$ is one of the bundles in the augmented Sankey definition described above. This graph is built according to Algorithm 1.

Appendix B. Querying flow values

The final step is to take the graph G_S describing the final structure of the diagram, and apply it to the underlying database to obtain actual values for the flows. The result is contained in another layered graph, the results graph $G_R = (V_R, E_R, L_R)$.

The database contains the flow table T_F (Figure 1). There may be a diagram-wide filter on material types or time intervals to be shown, which yields a subset of flows F'. The effect of filtering at this stage is different to filtering the flows selected by a bundle, since flows not in F' will not be implicitly added to ensure conservation of mass.

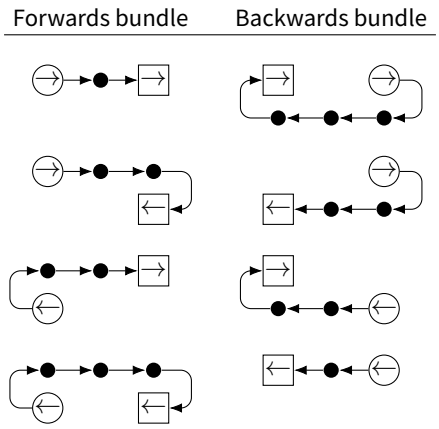


Figure 9: Dummy nodes (black dots) are needed to connect the source (circle) and target (square) of a bundle, if they are not in adjacent layers. There are eight possible combinations depending on the direction and relative placement of the source and target: for the situation drawn here, where the source and target are two layers apart, between one and three dummy nodes may be needed. This contrasts with a non-Sankey graph, where the nodes do not have a direction and only one dummy node would be needed in each case.

The flows belonging to a bundle b_{vw} are then selected by the function:

$$\text{select\_flows}(b_{vw}) = \left\{ f_{jkmt} \,\middle|\, f_{jkmt} \in F', \text{matches}(v, w, f_{jkmt}), m \in S_b^M, t \in S_b^T \right\} \tag{B.1}$$

where S_b^M is a selection of materials, S_b^T is a selection of time intervals, and

$$\text{matches}(v, w, f_{jkmt}) = \begin{cases} j \in S_v, & k \in S_w & & \text{if } v \neq \boxtimes, w \neq \boxtimes \\ j \notin S_w, & k \in S_w, & f_{jkmt} \notin F_{\text{used}} & \text{if } v = \boxtimes \\ j \in S_v, & k \notin S_v, & f_{jkmt} \notin F_{\text{used}} & \text{if } w = \boxtimes \end{cases} \tag{B.2}$$

Here $\boxtimes$, meaning "elsewhere", is a place-holder for processes which are not selected by any process groups. Note that this definition ensures that internal flows within a group are not included. F_{used} is the set of flows which have been selected by non-Elsewhere bundles:

$$F_{\text{used}} = \bigcup \text{select\_flows}(b_{vw}), \quad b_{vw} \in B, v \neq \boxtimes, w \neq \boxtimes \tag{B.3}$$

The final results graph has nodes

$$V_R = \{(v, l) \,|\, v \in V_S, l \in \text{labels}\,(P_v)\} \tag{B.4}$$

where labels (P_v) is the set of labels which are the co-domain of the partition function P_v assigned to group v. These nodes correspond directly to nodes in the Sankey diagram.

The results graph edges correspond to Sankey links, and are tuples $(j, k, m, t, x) \in E_R$, where j and k are the source and target in V_R, m is the material, t is the time interval, and x is the link value. The edges are found as

$$E_R = \bigcup_{(j,k,b_{vw}) \in E} \text{aggregate\_flows}\left(\text{select\_flows}(b_{vw}), \textit{measure}, P_j, P_k, P_b^M, P_b^T\right) \tag{B.5}$$

where *measure* is a place-holder for the measure of interest (often mass), P_j and P_k are the partitions defined by the node groups or waypoints at the start and end of the edge, P_b^M is the partition defined by the bundle b

for the level of detail of the final links, and P^T is a partition of time intervals. In most circumstances, it will be desirable to use the same time partition across the whole diagram, so P^T does not depend on the bundle b. Finally, the operation aggregate_flows() is expressed in pseudo-SQL as:

$$\text{aggregate\_flows}\left(\text{flows}, \text{measure}, P_{\text{source}}, P_{\text{target}}, P_{\text{material}}, P_{\text{time}}\right) = \tag{B.6}$$

```
SELECT source, target, material, time, SUM(measure) as value
FROM (
  SELECT P←source(flow) AS source,
         P→target(flow) AS target,
         Pmaterial(flow) AS material
         Ptime(flow) AS time
  FROM flows AS flow
)
GROUP BY source, target, material, time
```

A complication arises when aggregating flows using their source or target as they enter or leave a process. To get meaningful results, the targets of the incoming flows should usually be grouped in the same way as the sources of the outgoing flows. To allow this, a special dimension name called "process" is introduced: if a partition P_n refers to "process.location", say, then the partition can be used in two forms. $P_n^{\leftarrow}$ refers to "source.location", and is applied to outgoing flows, while $P_n^{\rightarrow}$ refers to "target.location", and is applied to incoming flows.

The ordering of the results graph L_R follows directly from the ordering of the previous graph L_S, given an ordering of the partition labels.

References

Alemasoom, H., Samavati, F., Brosz, J., Layzell, D., 2015. EnergyViz: An interactive system for visualization of energy systems. The Visual Computer 32, 403–413. doi:10.1007/s00371-015-1186-8.

Allwood, J., Ralph, D., Richards, K., Fenner, R., Linden, P., Dennis, J., Gilligan, C., Pyle, J., Kopec, G., Bajželj, B., Curmi, E., Qin, Y., Lupton, R., . Foreseer - integrated resource analysis. http://www.uselessgroup.org/foreseer-integrated-resource-analysis.

Bajželj, B., Allwood, J.M., Cullen, J.M., 2013. Designing Climate Change Mitigation Plans That Add Up. Environmental Science & Technology 47, 8062–8069. doi:10.1021/es400399h.

Brunner, P.H., Rechberger, H., 2003. Practical Handbook of Material Flow Analysis. CRC Press.

Cabrera Serrenho, A., Allwood, J.M., 2016. Material Stock Demographics: Cars in Great Britain. Environmental Science & Technology 50, 3002–3009. doi:10.1021/acs.est.5b05012.

Chaudhuri, S., Dayal, U., 1997. An Overview of Data Warehousing and OLAP Technology. SIGMOD Rec. 26, 65–74. doi:10.1145/248603.248616.

Chen, C., Yan, X., Zhu, F., Han, J., Yu, P.S., 2009. Graph OLAP: A multi-dimensional framework for graph data analysis. Knowledge and Information Systems 21, 41–63. doi:10.1007/s10115-009-0228-9.

Cullen, J.M., Allwood, J.M., 2010. The efficient use of energy: Tracing the global flow of energy from fuel to service. Energy Policy 38, 75–81. doi:10.1016/j.enpol.2009.08.054.

Cullen, J.M., Allwood, J.M., 2013. Mapping the Global Flow of Aluminum: From Liquid Aluminum to End-Use Goods. Environmental Science & Technology 47, 3057–3064. doi:10.1021/es304256s.

Cullen, J.M., Allwood, J.M., Bambach, M.D., 2012. Mapping the Global Flow of Steel: From Steelmaking to End-Use Goods. Environmental Science & Technology 46, 13048–13055. doi:10.1021/es302433p.

Curmi, E., Fenner, R., Richards, K., Allwood, J.M., Bajželj, B., Kopec, G.M., 2013. Visualising a Stochastic Model of Californian Water Resources Using Sankey Diagrams. Water Resources Management 27, 3035–3050. doi:10.1007/s11269-013-0331-2.

Density Design, 2010. Fineo. http://www.densitydesign.org/research/fineo/.

Diener, D.L., Tillman, A.M., 2016. Scrapping steel components for recycling—Isn't that good enough? Seeking improvements in automotive component end-of-life. Resources, Conservation and Recycling 110, 48–60. doi:10.1016/j.resconrec.2016.03.001.

Flint, I., Allwood, J.M., Cabrera Serrenho, A., Lupton, R.C., in preparation. More than just 'steel' - characterising flows of flat steel in the EU by grade, thickness and coating .

Gansner, E.R., Koutsofios, E., North, S.C., Vo, G.P., 1993. A technique for drawing directed graphs. Software Engineering, IEEE Transactions on 19, 214–230.

Geyer, R., Davis, J., Ley, J., He, J., Clift, R., Kwan, A., Sansom, M., Jackson, T., 2007. Time-dependent material flow analysis of iron and steel in the UK: Part 1: Production and consumption trends 1970–2000. Resources, Conservation and Recycling 51, 101–117. doi:10.1016/j.resconrec.2006.08.006.

Giuffrida, A., Romano, M.C., Lozza, G., 2011. Thermodynamic analysis of air-blown gasification for IGCC applications. Applied Energy 88, 3949–3958. doi:10.1016/j.apenergy.2011.04.009.

Gonzalez Hernandez, A., Lupton, R.C., Williams, C., Cullen, J.M., 2017. From control data to real-time maps of material and energy consumption: The example of a steel-making plant, in: Energy Procedia, Cardiff, UK.

ifu Hamburg, 2017. E!Sankey software. `https://www.ifu.com/en/e-sankey/`.

Kalt, G., 2015. Biomass streams in Austria: Drawing a complete picture of biogenic material flows within the national economy. Resources, Conservation and Recycling 95, 100–111. doi:10.1016/j.resconrec.2014.12.006.

Kimball, R., 1996. The Data Warehouse Toolkit: Practical Techniques for Building Dimensional Data Warehouses. John Wiley & Sons, Inc., New York, NY, USA.

Konadu, D.D., Mourão, Z.S., Allwood, J.M., Richards, K.S., Kopec, G., McMahon, R., Fenner, R., 2015. Land use implications of future energy system trajectories—The case of the UK 2050 Carbon Plan. Energy Policy 86, 328–337. doi:10.1016/j.enpol.2015.07.008.

Kopec, G.M., 2015. Examining Natural Resource Futures with Material Flow Analysis. Ph.D. thesis. University of Cambridge.

Kosara, R., Bendix, F., Hauser, H., 2006. Parallel sets: Interactive exploration and visual analysis of categorical data. Visualization and Computer Graphics, IEEE Transactions on 12, 558–568.

Leal-Ayala, D.R., Allwood, J.M., Petavratzi, E., Brown, T.J., Gunn, G., 2015. Mapping the global flow of tungsten to identify key material efficiency and supply security opportunities. Resources, Conservation and Recycling 103, 19–28. doi:10.1016/j.resconrec.2015.07.003.

Löfving, E., Grimvall, A., Palm, V., 2006. Data cubes and matrix formulae for convenient handling of physical flow data. Journal of Industrial Ecology 10, 43–60.

Lupton, R.C., 2016a. D3-sankey-diagram. GitHub: ricklupton/d3-sankey-diagram doi:10.5281/zenodo.162331.

Lupton, R.C., 2016b. Sankeyview. GitHub: ricklupton/sankeyview doi:10.5281/zenodo.161970.

Lupton, R.C., Allwood, J.M., 2016. Research data supporting "Visual analyses of multidimensional data for understanding resource use" doi:10.17863/CAM.6038.

Lupton, R.C., Allwood, J.M., under review. Incremental Material Flow Analysis with Bayesian Inference .

Möller, A., Page, B., Rolf, A., Wohlgemuth, V., 2000. Foundations and applications of computer based material flow networks for environmental management, in: Environmental Information Systems in Industry and Public Administration. Idea Group Inc (IGI).

Mourão, Z.S., Konadu, D.D., Lupton, R.C., Allwood, J.M., in preparation. An investigation of the potential for NOx and PM2.5 emissions reductions at local and national level in the UK: Synergies and conflicts with climate change goals .

Narasimhan, S., Jordache, C., 1999. 1 - The Importance of Data Reconciliation and Gross Error Detection, in: Data Reconciliation and Gross Error Detection. Gulf Professional Publishing, Burlington, pp. 1–31.

Nuttbohm, K., Fischer, L., Muckenfuss, L., Baiboks, S., 2009. Visualising sustainability communication with Sankey diagrams – a viable approach?, in: Environmental Informatics and Industrial Environmental Protection: Concepts, Methods and Tools, Berlin.

Paoli, L., Lupton, R.C., Cullen, J.M., 2017. Conversion device energy consumption: A practical methodology and uncertainty analysis, in: Energy Procedia, Cardiff, UK.

Pedersen, T.B., Jensen, C.S., Dyreson, C.E., 2001. A Foundation for Capturing and Querying Complex Multidimensional Data.

Riehmann, P., Hanfler, M., Froehlich, B., 2005. Interactive sankey diagrams, in: Information Visualization, 2005. INFOVIS 2005. IEEE Symposium On, IEEE. pp. 233–240.

Schmidt, M., 2008a. The Sankey Diagram in Energy and Material Flow Management: Part I: History. Journal of Industrial Ecology 12, 82–94. doi:10.1111/j.1530-9290.2008.00004.x.

Schmidt, M., 2008b. The Sankey Diagram in Energy and Material Flow Management: Part II: Methodology and Current Applications. Journal of Industrial Ecology 12, 173–185. doi:10.1111/j.1530-9290.2008.00015.x.

Soundararajan, K., Ho, H.K., Su, B., 2014. Sankey diagram framework for energy and exergy flows. Applied Energy 136, 1035–1042. doi:10.1016/j.apenergy.2014.08.070.

Sugiyama, K., Tagawa, S., Toda, M., 1981. Methods for Visual Understanding of Hierarchical System Structures. IEEE Transactions on Systems, Man and Cybernetics 11, 109–125. doi:10.1109/TSMC.1981.4308636.